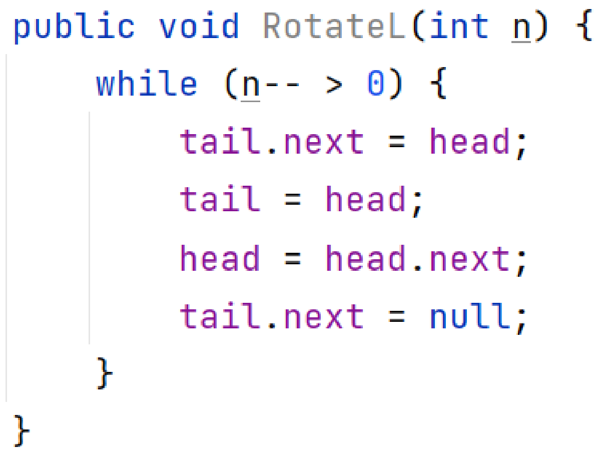
أسئلة على اللنكد ليست

شعبة 3

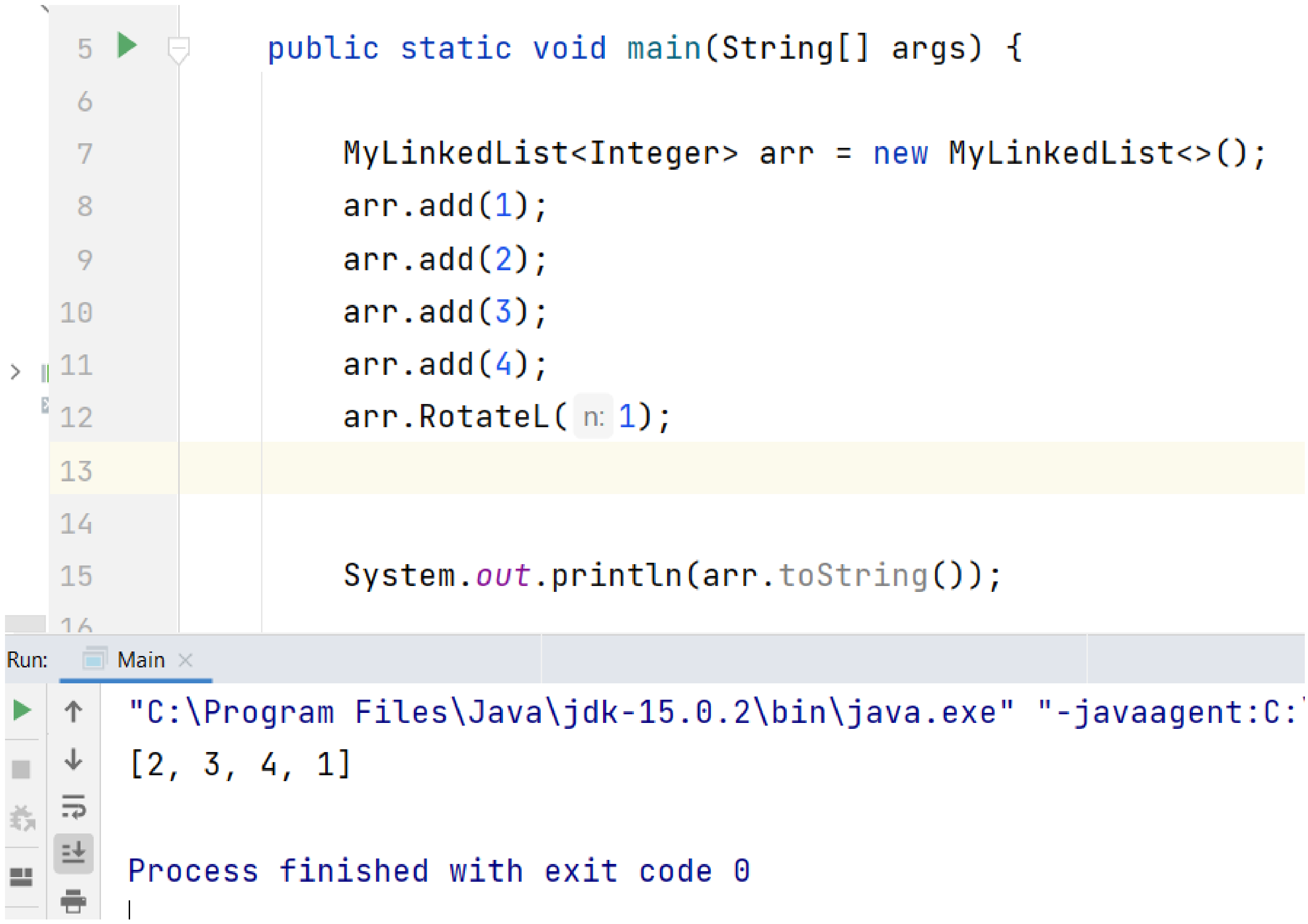
محمد ابو صفط

1* ميثود داخل كلاس اللنكد ليست يعمل روتيت لليسار بمقدار ن :



```java
public void RotateL(int n) {
    while (n-- > 0) {
        tail.next = head;
        tail = head;
        head = head.next;
        tail.next = null;
    }
}
```

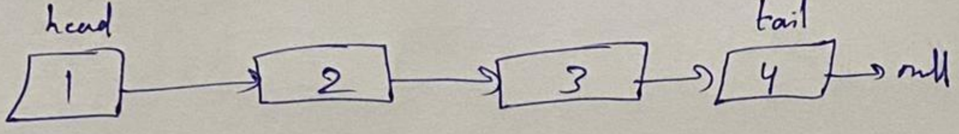
Run method:



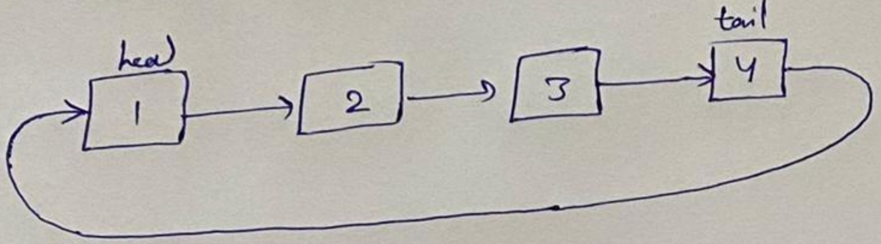
```java
    public static void main(String[] args) {

        MyLinkedList<Integer> arr = new MyLinkedList<>();
        arr.add(1);
        arr.add(2);
        arr.add(3);
        arr.add(4);
        arr.RotateL( n: 1);


        System.out.println(arr.toString());
```

```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaagent:C:
[2, 3, 4, 1]

Process finished with exit code 0
```

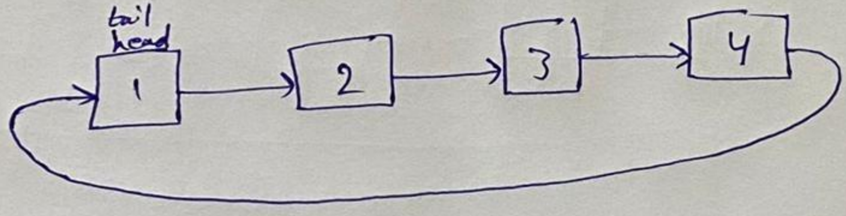
* Rotate Left



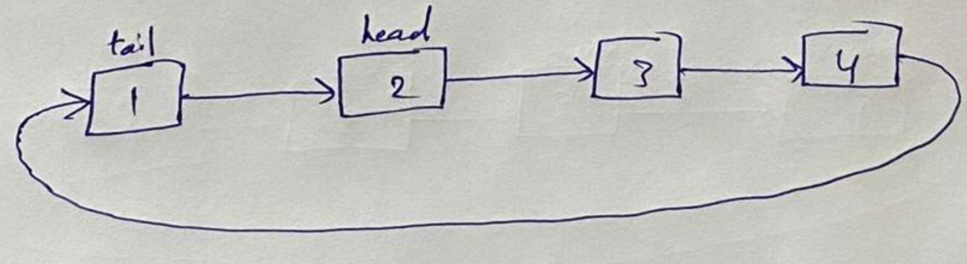
=> tail.next = head ;



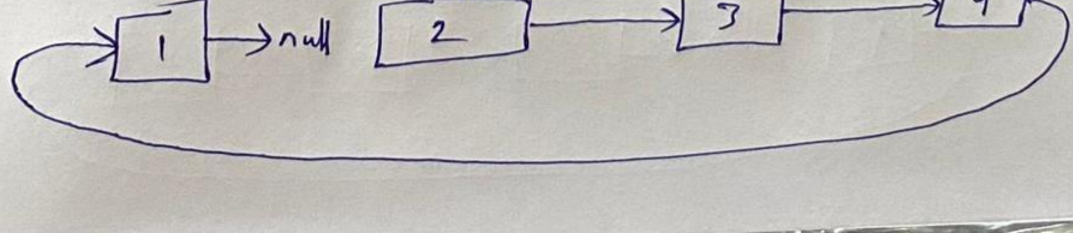
=> tail = head ;



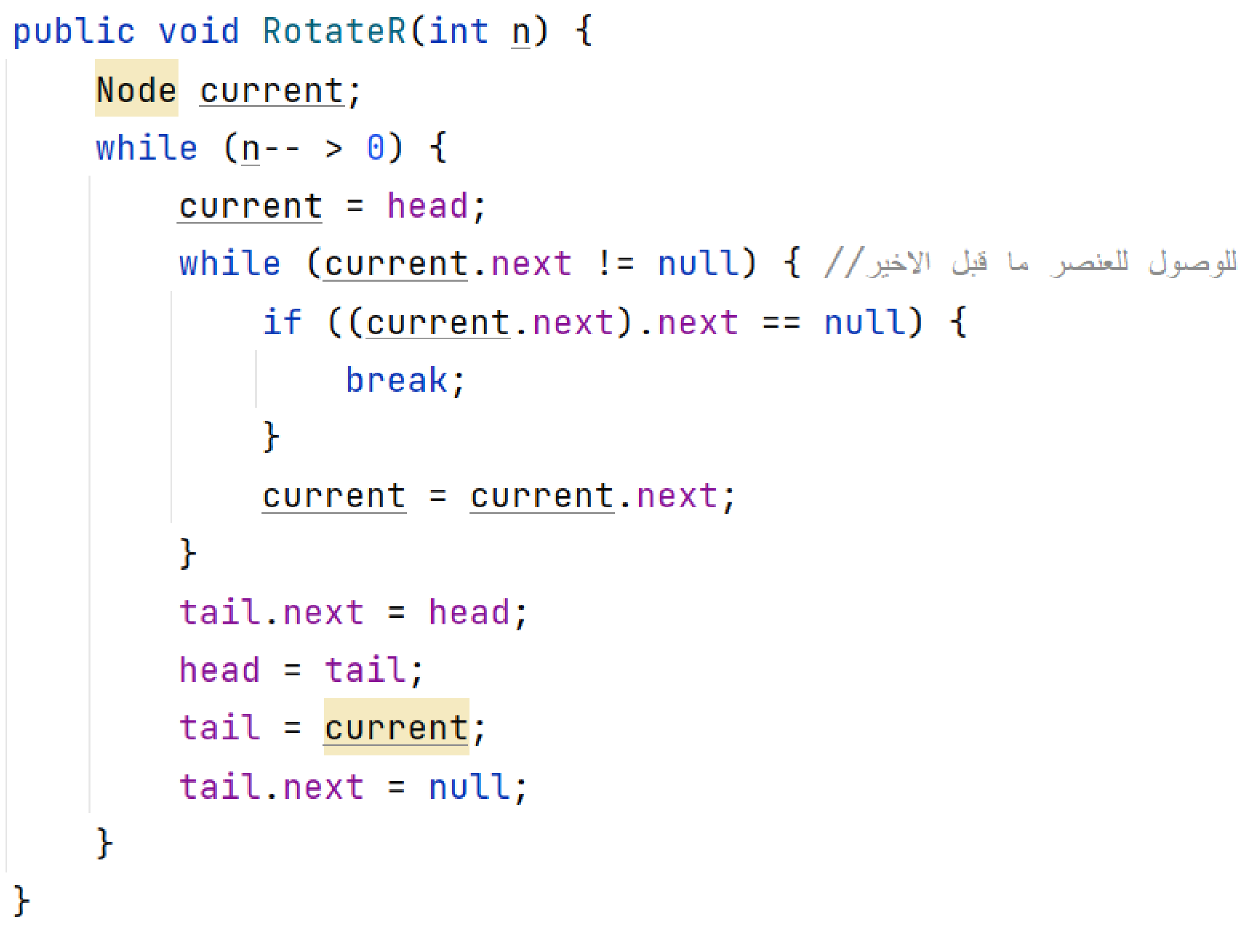
=> head = head.next ;



=> tail.next = null ;

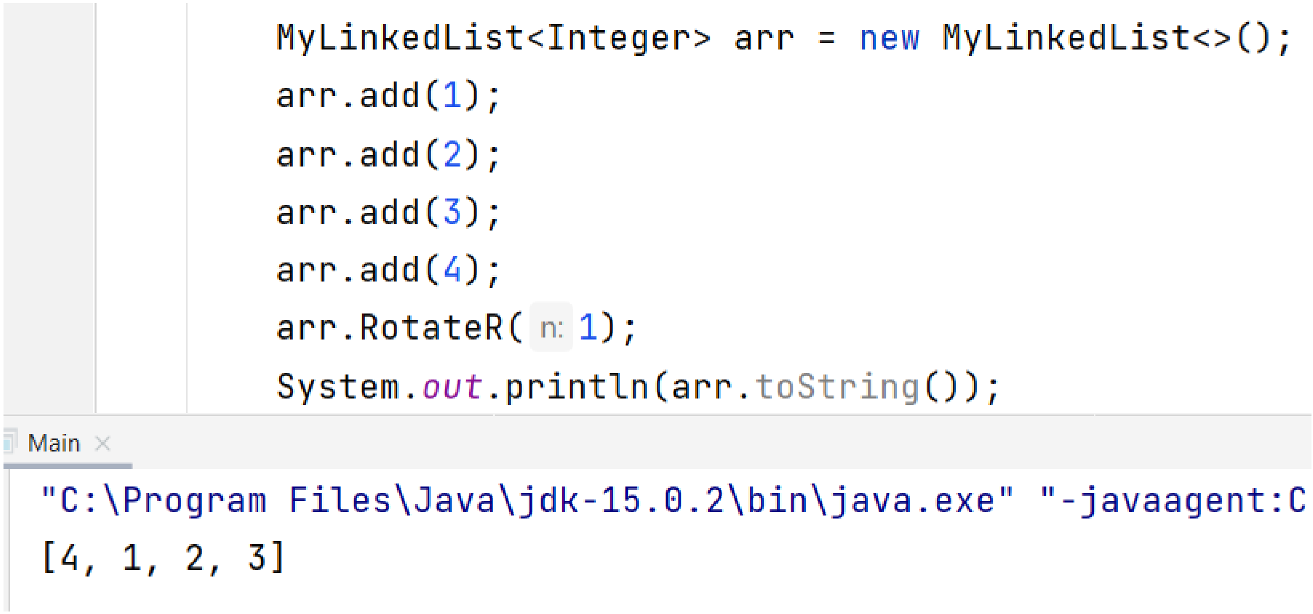


2* ميثود داخل كلاس اللنكد ليست يعمل روتيت لليمين بمقدار ن :



```java
public void RotateR(int n) {
    Node current;
    while (n-- > 0) {
        current = head;
        while (current.next != null) { //الوصول للعنصر ما قبل الاخير
            if ((current.next).next == null) {
                break;
            }
            current = current.next;
        }
        tail.next = head;
        head = tail;
        tail = current;
        tail.next = null;
    }
}
```

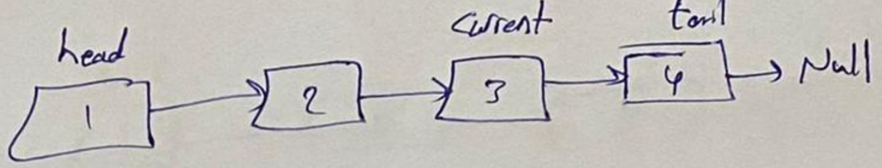
Run method:



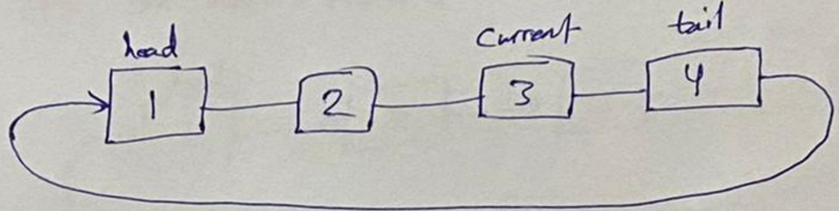
```java
        MyLinkedList<Integer> arr = new MyLinkedList<>();
        arr.add(1);
        arr.add(2);
        arr.add(3);
        arr.add(4);
        arr.RotateR( n: 1);
        System.out.println(arr.toString());
```

```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaagent:C
[4, 1, 2, 3]
```

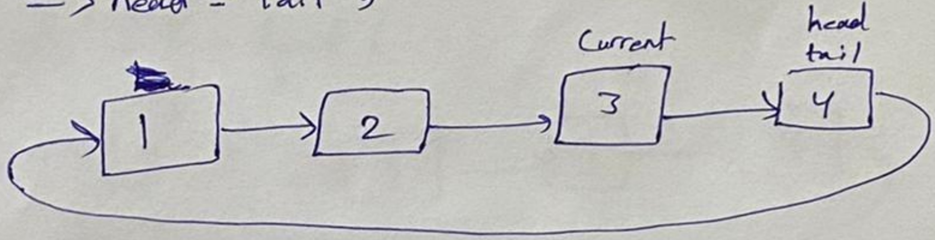
* Rotate Right.



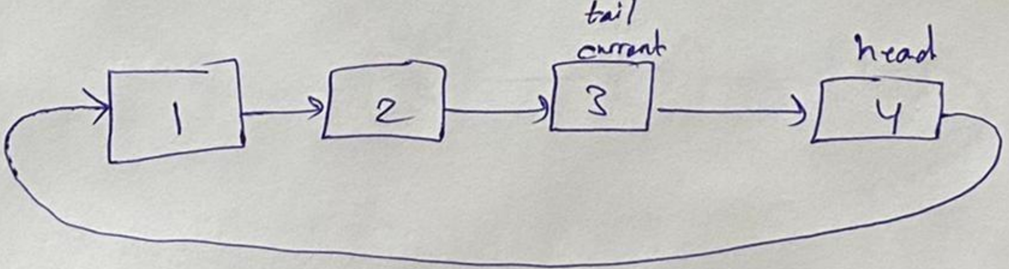
⟹ tail.next = head ;



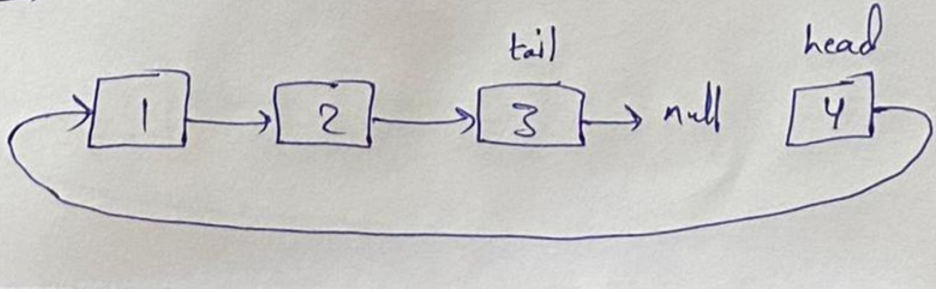
⟹ head = tail ;



⟹ tail = current ;



⟹ tail.next = null ;



*3

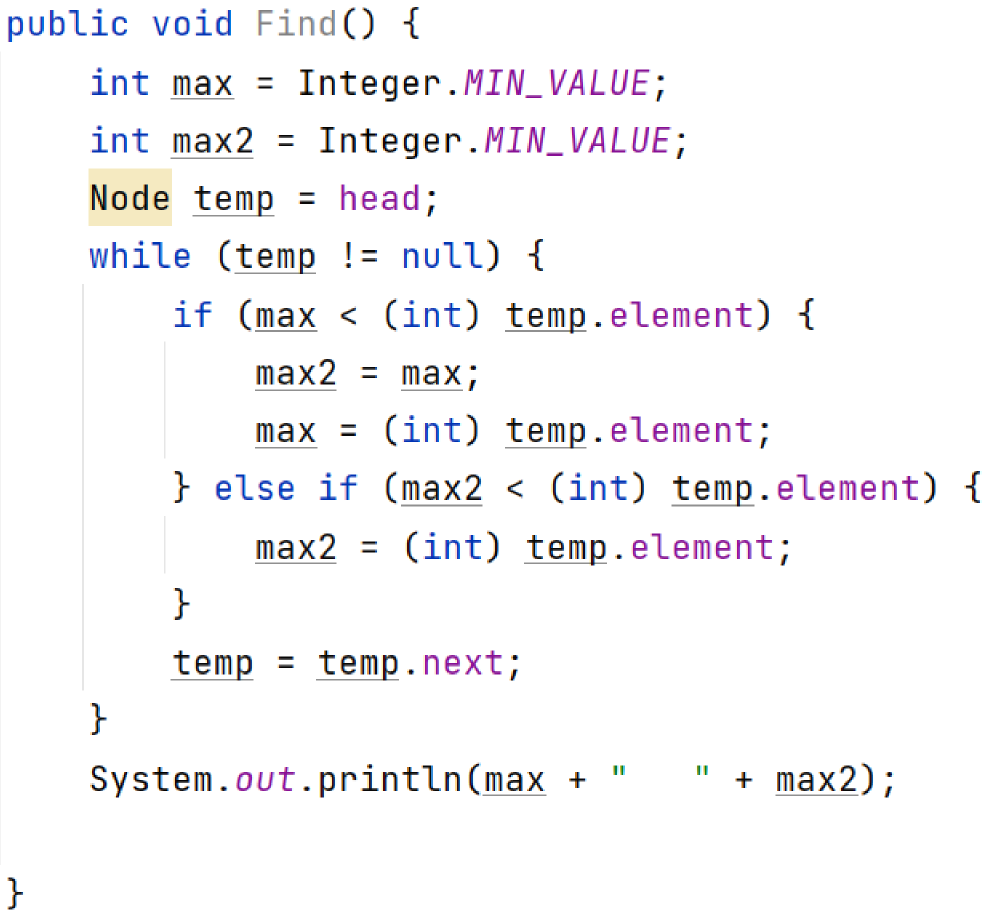
Using your own LinkedList Class ”MyLinkedList”

<u>Note</u>:

*Create a main method and create instance of MyLinkedList class and call the methods in a,b.

*Attach a photo of the output for a,b.(There will be some marks on photo)

a.Implement a method **to find maximum and second maximum** element of integer LinkedList (8marks)



```
public void Find() {
    int max = Integer.MIN_VALUE;
    int max2 = Integer.MIN_VALUE;
    Node temp = head;
    while (temp != null) {
        if (max < (int) temp.element) {
            max2 = max;
            max = (int) temp.element;
        } else if (max2 < (int) temp.element) {
            max2 = (int) temp.element;
        }
        temp = temp.next;
    }
    System.out.println(max + "   " + max2);

}
```

*4

the following method accepts 2 nodes, each one is a head of a linked list of type Integer, both lists have the same length. each list represents a non-negative number,

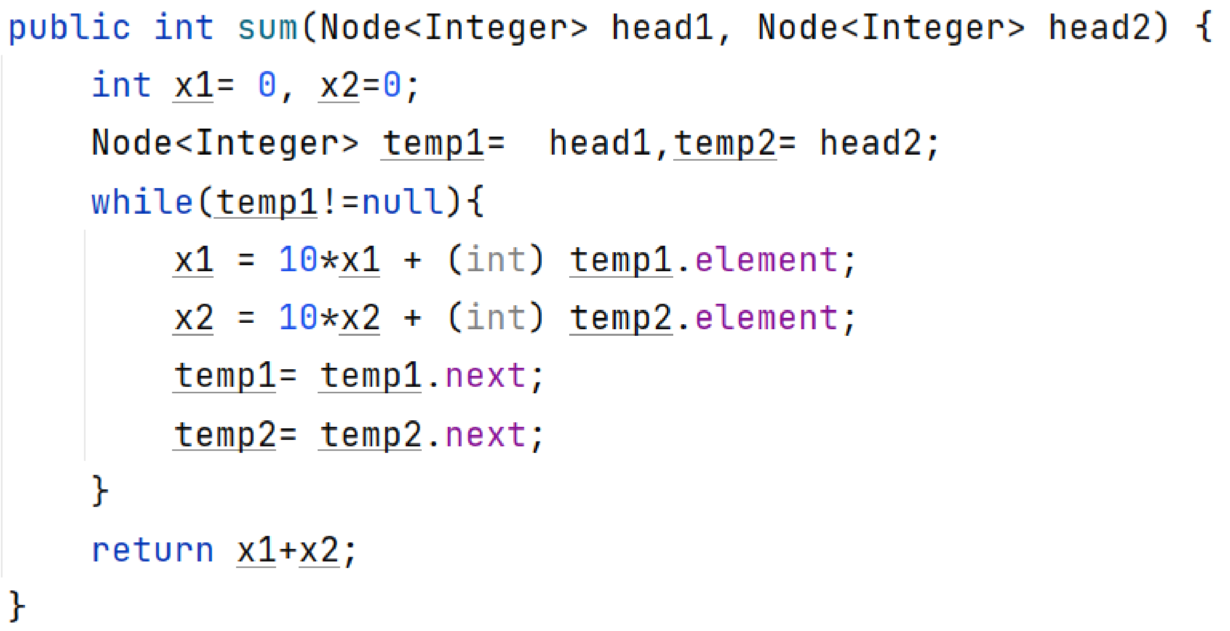
such that each digit of the number is contained in a node, the right node (last node) is the ones place (خانة الآحاد)

your task is to calculate the sum of the two numbers.

use the following method header:

public int sum(Node<Integer> head1, Node<Integer> head2) { ... }

solution:



```
public int sum(Node<Integer> head1, Node<Integer> head2) {
    int x1= 0, x2=0;
    Node<Integer> temp1=  head1,temp2= head2;
    while(temp1!=null){
        x1 = 10*x1 + (int) temp1.element;
        x2 = 10*x2 + (int) temp2.element;
        temp1= temp1.next;
        temp2= temp2.next;
    }
    return x1+x2;
}
```

*5

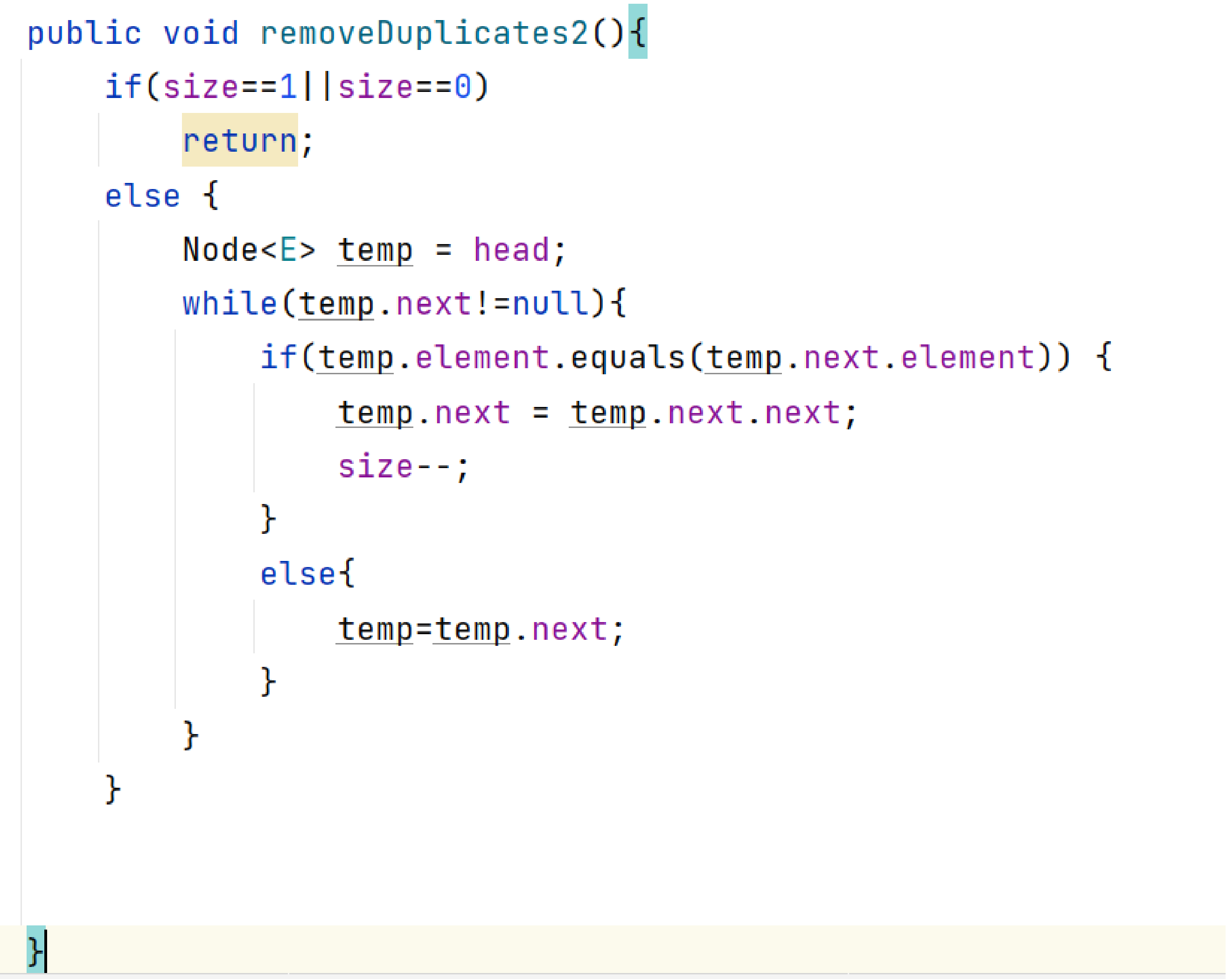
implement a method inside MyLinkedList class to remove the duplicates in the list with a max complexty of O(n),

assume it is called only when the list is sorted.

use the following header:

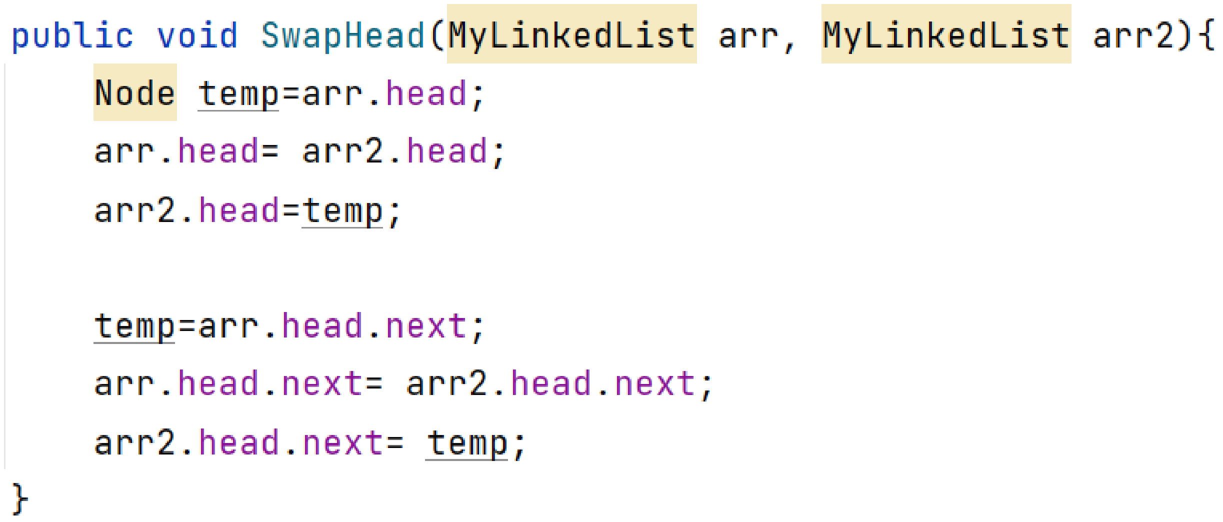
public void removeDuplicates() { }

Solution:



```java
public void removeDuplicates2(){
    if(size==1||size==0)
        return;
    else {
        Node<E> temp = head;
        while(temp.next!=null){
            if(temp.element.equals(temp.next.element)) {
                temp.next = temp.next.next;
                size--;
            }
            else{
                temp=temp.next;
            }
        }
    }


}
```

6* ميثود يستقبل 2 لنكد ليست ويعمل سواب لأول عنصر بكلا اللنكد ليست



```java
public void SwapHead(MyLinkedList arr, MyLinkedList arr2){
    Node temp=arr.head;
    arr.head= arr2.head;
    arr2.head=temp;

    temp=arr.head.next;
    arr.head.next= arr2.head.next;
    arr2.head.next= temp;
}
```

*7

implement a method inside MyLinkedList class to swap two nodes specified at indecies index1 and index2

you are NOT allowed to swap values inside the nodes, you should swap the whole nodes.

use the following method header:

public void swap(int index1, int index2) { ... }

test case1:

list before swapping: [1, 2, 3, 4, 5]

list after swapping node at index=1 with node at index=3 : [1, 4, 3, 2, 5]

solution

```java
public void swapNodes(int index1, int index2) {
    if(!(index1<size&&index2<size&&index1>=0&&index2>=0)){
        System.out.println("error");
        return;
    }
    if ((head == tail) || index1==index2)
        return;

    if(index1>index2){
        int temp = index1;
        index1 = index2;
        index2 = temp;
    }
    Node prev1=null, node1=head, next1,
            prev2=null, node2=head, next2;

    for (int i = 0; i < index1; i++) {
        prev1 = node1;
        node1 = node1.next;
    }
    next1  =node1.next;

    for (int i = 0; i < index2; i++) {
```

```
    prev2 = node2;
    node2 = node2.next;
}
next2  =node2.next;
//----------Start Swaping---------
node1.next= next2;
if(index1!=0)
    prev1.next = node2;

if(index2!= index1+1)//(prev2!=node1)
prev2.next= node1;

if(prev2!=node1) // (index2!= index1+1)
node2.next = next1;
else {
    node2.next = node1;
}
if(index1== 0)
    head=node2;
if(index2 == size-1)
    tail=node1;

}
```

*8

Using your own LinkedList Class,

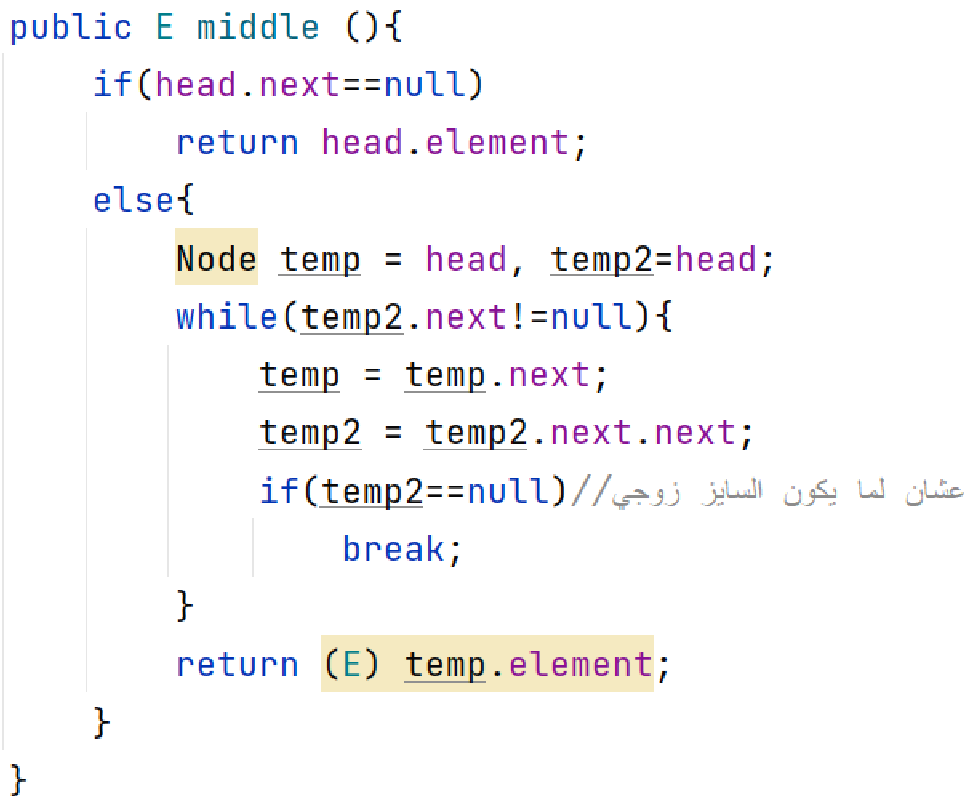
a. Write a method to find the middle element in LinkedList **without using size(), get() methods** (8marks).

The signature of the method:

public E middle (){

}

Solution:



```
public E middle (){
    if(head.next==null)
        return head.element;
    else{
        Node temp = head, temp2=head;
        while(temp2.next!=null){
            temp = temp.next;
            temp2 = temp2.next.next;
            if(temp2==null)//عشان لما يكون السايز زوجي
                break;
        }
        return (E) temp.element;
    }
}
```