**Set A**

| **Python** | **Java** |
| --- | --- |
| def oddRev( head ):  newH = None  n = head  while n!=None:  if( n.elem%2==1 ):  if( newH == None ):  newH = Node(n.elem)  else:  newN = Node(n.elem)  newN.next = newH  newH = newN  n=n.next  return newH | public static Node oddRev( Node head ){  Node newH = null;  Node n = head;  while (n!=null){  if ( n.elem%2==1 ){  if ( newH == null )  newH = new Node(n.elem);  else{  Node newN = new Node(n.elem);  newN.next = newH;  newH = newN;  }  }  n=n.next;  }  return newH;  } |

**Set B**

| **Python** | **Java** |
| --- | --- |
| def evenRev( head ):  newH = None  n = head  while n!=None:  if( n.elem%2==0 ):  if( newH == None ):  newH = Node(n.elem)  else:  newN = Node(n.elem)  newN.next = newH  newH = newN  n=n.next  return newH | public static Node evenRev( Node head ){  Node newH = null;  Node n = head;  while (n!=null){  if ( n.elem%2==0 ){  if ( newH == null )  newH = new Node(n.elem);  else{  Node newN = new Node(n.elem);  newN.next = newH;  newH = newN;  }  }  n=n.next;  }  return newH;  } |

**RUBRIC**

|  | **Criteria** | **Marks** |
| --- | --- | --- |
| **1** | **Properly declaring method/function using proper parameter** | **2** |
| **2** | **Properly looping through the given linked list** | **2** |
| **3** | **Creating the newHead node** | **2** |
| **4** | **Checking even/odd and Creating new nodes** | **3** |
| **5** | **Properly connecting the nodes** | **3** |
| **6** | **Updating the newHead** | **2** |
| **7** | **Returning the newHead** | **1** |