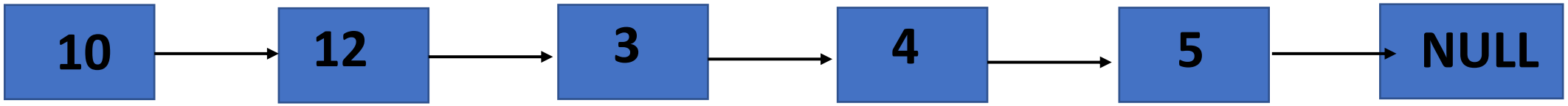

FACE
Prep

Segregate even and odd nodes in a Linked List

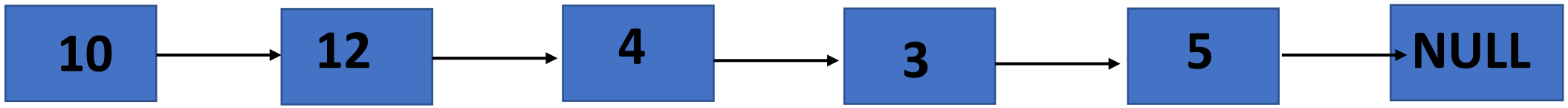
Segregate even and odd nodes in a Linked List

Problem: Given a linked list, Segregate even and odd nodes in a Linked List.

Sample Input:



Sample Output:



Solutions

- In this problem we need to sort the linked list, so that all the even nodes will be arranged before odd nodes.

- Order should be maintain.

Approach 1:

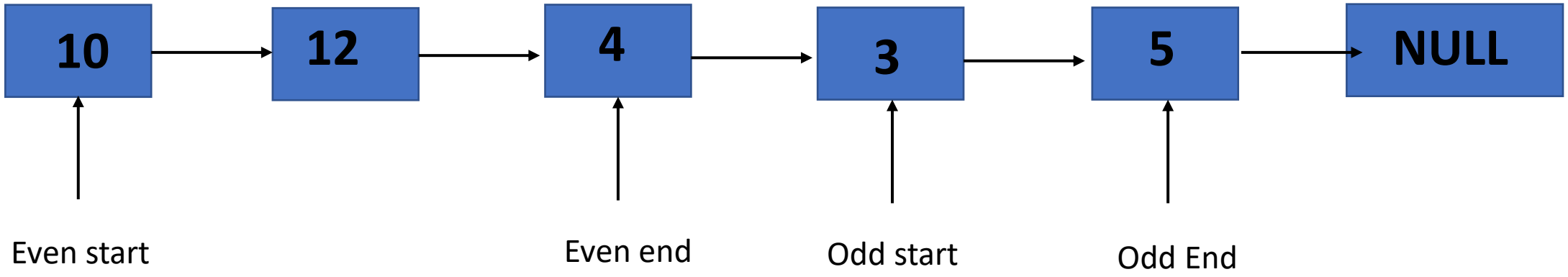
- Delete odd node from the beginning and insert in to last.

- Problem with this approach is we need to use two traversal for finding the last node of the list and deletion of the last node.

Approach 2:

- Traverse the linked list and connect all the even nodes in one list and connect all the odd nodes in once list.

- And then connect even list last node and odd list first node.



```
1   import java.util.*;
2   class Main {
3         Node head;
4         class Node {
5               int data;
6               Node next;
7               Node( int d) {
8                     data =d;
9                     next=null;
10              }
11        }
12        void  segregateEvenOdd() {
13              Node evenStart = null;
14              Node evenEnd = null;
15              Node oddStart = null;
16              Node oddEnd = null;
17              Node currentNode  = head;
18
19
20
21
22
```

```
1       while (currentNode != null) {
2             int element = currentNode.data;
3             if(element %2 == 0) {
4                   if(evenStart == null) {
5                         evenStart = currentNode;
6                         evenEnd = evenStart;
7                   }
8                   else {
9                         evenEnd.next = currentNode;
10                        evenEnd = evenEnd.next;
11                  }
12            }
13            else {
14                  if(oddStart == null) {
15                        oddStart = currentNode;
16                        oddEnd = oddStart;
17                  }
18
19
20
21
22
```

```
            else {
                    oddEnd.next = currentNode;
                    oddEnd = oddEnd.next;
                }
        }
        currentNode = currentNode.next;
    }
    if(oddStart == null || evenStart == null) {
        return ;
    }
    evenEnd.next = oddStart;
    oddEnd.next=null;
    head = evenStart;
    }
```

```
1       void push(int new_data) {
2           Node new_node = new Node(new_data);
3
4           new_node.next = head;
5           head = new_node;
6       }
7
8       void printList() {
9           Node temp = head;
10          while(temp !=null) {
11
12                  System.out.print(temp.data+" ");
13                  temp = temp.next;
14          }
15
16          System.out.print();
17      }
18
19
20
21
22
```

```
1       public static void main(Stirng args[]) {
2               Main main = new Main();
3               Scanner sc=new Scanner(System.in);
4               int n=sc.nextInt();
5               for(int i=0;i<n;i++) {
6                       int m=sc.nextInt();
7                       main.push(m);
8               }
9               main.segregateEvenOdd();
10              main.printList();
11          }
12  }
13
14
15
16
17
18
19
20
21
22
```

THANK YOU

FACE Prep