ODD EVEN LINKED LIST

✡ In this problem, we are given a singly linked list and our job is to return a list where all the odd indices are at the start and even indices are at the end.

Brute force solution is to create a desired format of our linked list in an array after one iteration. Then in another iteration we just change values in linkedlist Time Complexity is O(2N) and space complexity is O(N).

Optimal Solution will eliminate use of array. We use 2 pointers, odd (head) and even (head → next). At each step we will modify links of each node.

C++ :

```
Node * oddEvenLinkedList(Node* head) {
    Node * odd = head;
    Node * even = head → next;
    Node * evenHead = head → next;
    while (even != nullptr && even.next
                                != nullptr){
        odd → next = odd → next → next;
        even → next = even → next → next;
        odd = odd → next;
        even = even → next;
```

```
    }
    odd -> next = evenHead ;
    return head ;
}
}
```