Date ____________ Page ______

Day -123

Linked List - 9

* **Add 2 Numben:**

First → 9 7 6 8 4

Second → 6 4 3 8

⇒ We have to add no. in this way -

$$\begin{array}{r}
9\ 7\ 6\ 8\ 4 \\
+\ 6\ 4\ 3\ 8 \\
\hline
10\ 4\ 1\ 2\ 2 \\
\hline
\end{array}$$

⇒ Now, we have to return the answer in the form of linked list.

head → 1 → 0 → 4 → 1 → 2 → 2

⇒ First we reverse the LL -

head1 → $\overset{\text{curr1}}{4}$ → 8 → 6 → 7 → 9

head2 → $\overset{\text{curr2}}{8}$ → 3 → 4 → 6

⇒ Now, we will ~~start~~ stant adding the no.

head

=> → 2 → 2 → 1 → 4 → 0 → 1

=> Now reverse the above LL.

=> → 1 → 0 → 4 → 1 → 2 → 2

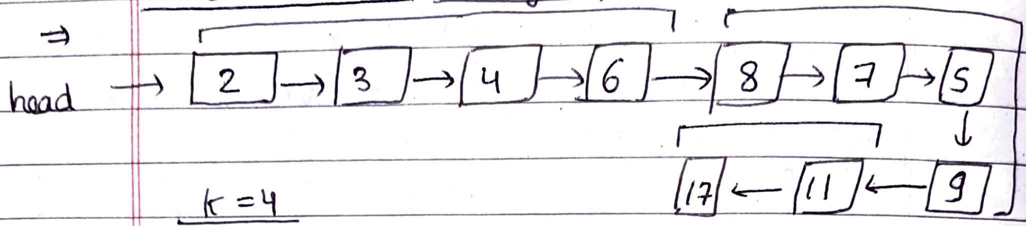
Code

```
Node *  reverse( Node * curr, Node * prev){
        if(curr == NULL)
                return prev;
        Node * front = curr → next;
        curr → next = prev;
    return reverse ( front, curr);
}

int main(){
        first = reverse (first, NULL);
        second = reverse ( ~~&~~ second, NULL);

        Node  * curr1 = first, * curr2 = second;
        Node * head = ~~NULL,~~ new Node (0); int carry =0;
        Node * tail = ~~NULL,~~ head;
        while( curr1 && curr2){
                int sum = curr1 → data +
                    curr2 → data + carry;
            {
```

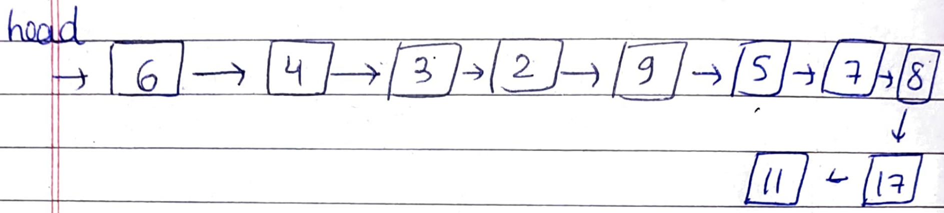
```
            Tail→next = new Node( sum % 10);
            Tail = Tail→next;
            curr1 = curr1→next;
            curr2 = curr2→next;
        ~~}~~ carry = sum / 10;
        }
        while (curr1){
            ~~Tail~~
            int sum = curr1→data + carry;
            Tail →next = new Node (sum % 10);
            Tail = Tail → next;
            curr1 = curr1 →next;
            carry = sum /10;
        }
        while( curr2){
            int sum = curr2 →data + carry;
            Tail →next = new Node (sum %10);
            Tail = Tail →next;
            curr2 = curr2 →next;
            carry = sum / 10;
        }
        while(carry){
            Tail →next = new Node( carry %10);
            carry /=10;
        }
        head = reverse (head→next , NULL);
        return head;
    }
```

* Reverse a LL ingroup of sizes :

⇒

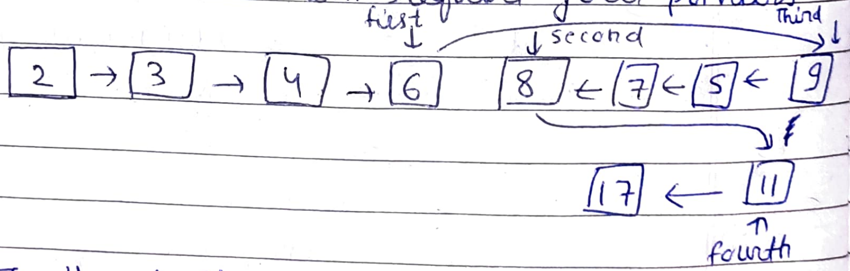


k = 4



⇒ To understand the approach, first we will understand how can I reverse 8 → 7 → 5 → 9 i.e. 9 → 5 → 7 → 8.

⇒ So,
for this we will require four pointers —



⇒ In the starting, we will have first and second only.

⇒ So, we will require prev, curr & front pointer to reverse the array.

⇒ Now, we will do reverse the LL procedure.

⇒ In the end or after reversing 4 nodes, prev becomes third and curr becomes fourth.

Date ____________ Page ____________

⇒ So, now,

```
        first → next = prev;
        second → next = curr;
```

⇒ So, code for only reversing that part —

Code

```
int x = k;
Node * second = first → next;
Node * prev = first, * front = NULL;
Node * curr = first → next;
while(x && curr){
        front = curr → next;
        curr → next = prev;
        prev = curr;
        curr = front;
        x--;
}
first → next = prev;
second → next = curr;
first = second;
```

⇒ Now, we will use this whole code to solve our problem.

⇒ Before that we, will create a dummy node.

```
Node * first = new Node(0);
first → next = head;
head = first;
while( first → next ){
        // Above code
};   first = head;                   delete first;
     head = head → next;             return head;
```