Addition of two linked list:

list ① → ⑨ → ⑧ → ② → ③ → null

list ② → ④ → ⑦ → ① → null

① → Reverse list ①

② → Reverse list ②

list① ③ → ② → ⑧ → ⑨ → null

out② ① → ⑦ → ④ → null

carry = 0

ival = 3
i = i·next,
jval = 1
j = j·next
sum = ival + jval + carry

val = sum % 10
carry = sum / 10

res·add list (val);

carry = $\cancel{0}$ $\cancel{1}$ 0
ival = $\cancel{3}$ $\cancel{2}$ $\cancel{8}$ $\cancel{9}$ 0
jval = $\cancel{1}$ $\cancel{7}$ $\cancel{4}$ 0
sum = $\cancel{4}$ $\cancel{9}$ $\cancel{12}$ $\cancel{10}$ 1
val = $\cancel{4}$ $\cancel{9}$ $\cancel{2}$ $\cancel{0}$ 1

linkedlist

① → ⑥ → ② → ⑨ → ④]

list1 : (9) — (4) — (3) — (6) |~

list2 : (8) — (7) — (3) |⌐

list1 → (6) — (3) — (4) — (9) → null()

i ↗

list2 → (3) — (7) — (8) → null

j ↗

carry = 0̸ 1̸ 1̸ 1̸ 0

ival = 6̸ 8̸ 4̸ 9̸ 0

jval = 8̸ 7̸ 8̸ 0̸ 0

sum = 8̸ 10̸ 13̸ 10̸ 1

val = 8̸ 0̸ 3̸ 0̸ 1

result → (1) → (0) → (3) → (0) → (9)

```
Node head1 = one.head;
Node head2 = two.head;
// don't change original list
// 1. reverse
head1 = reversePointer(head1);
head2 = reversePointer(head2);

// 2. add
Node i = head1;
Node j = head2;
linkedlist res = new linkedlist();
int carry = 0;
while(i != null || j != null || carry != 0) {
    int ival = i == null ? 0 : i.data;
    i = i == null ? null : i.next;

    int jval = j == null ? 0 : j.data;
    j = j == null ? null : j.next;

    int sum = ival + jval + carry;

    int val = sum % 10;
    carry = sum / 10;
    res.addFirst(val);
}

// 3. make original list again, reverse head1 and head2
head1 = reversePointer(head1);
head2 = reversePointer(head2);
// 4. return result
return res;
```
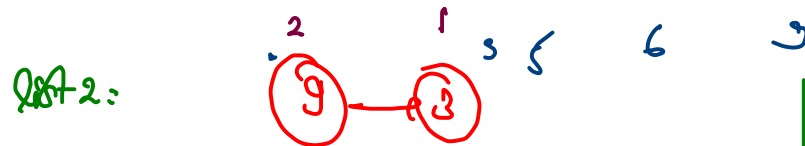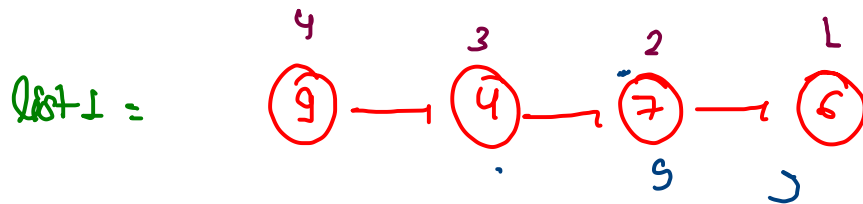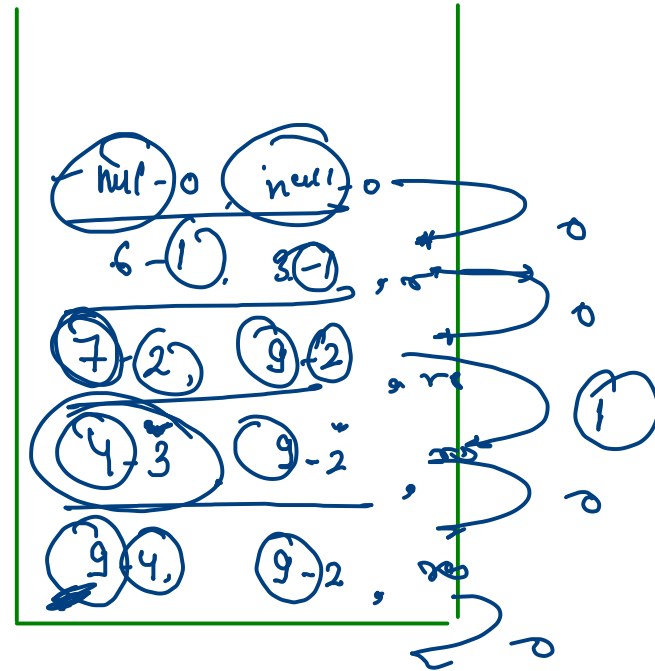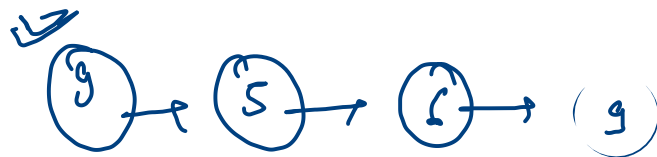
list1 =

4 → 3 → 2 → L

(9) — (4) — (7) — (6)

9)

2   1   3 5   6   9

list2 :

(9) — (3)

return type → carry :

res = new LinkedList<>()

(9) → (5) → (6) → (9)

null - 0   null - 0   0

6 - (1)   3 - (1)   0

(7) (2)   (9) (2)   1

(4) 3   (9) - 2   0

(9) (4)   (9) - 2   0

main

i1 == i2

i1 > i2

i2 > i1

Intersection of Linked List

time → $O(n) \times 3 =$
space → $O(1)$

temp1

head1 (a) → (b) → (c) → (d) → (e) → (f)

Size = 6

head2 (1) → (2) → (3) → (4) → (5)

Size = 8

temp2

(1) → (2) → (3)

(1) → (2) → (3) → (4)

Logic

— DRY

— Null

o/p

+

loop

Equ

Copy Linked List with Random pointer

① do not change actual linkedlist

Clone→

Neurons-
Active

```
Node {
  int data;
  Node next;
  Node random;
}
```

Even + even = Even.
Odd + odd = Even

80   40   null   10

10 → 20 → 30 → 40 → null   temp.

10'   20'   30'   40'
    30'   40'   null   10'

while (temp != null) } {

   temp.next.random = temp.random == null ? null) ; temp.
                                                       random.
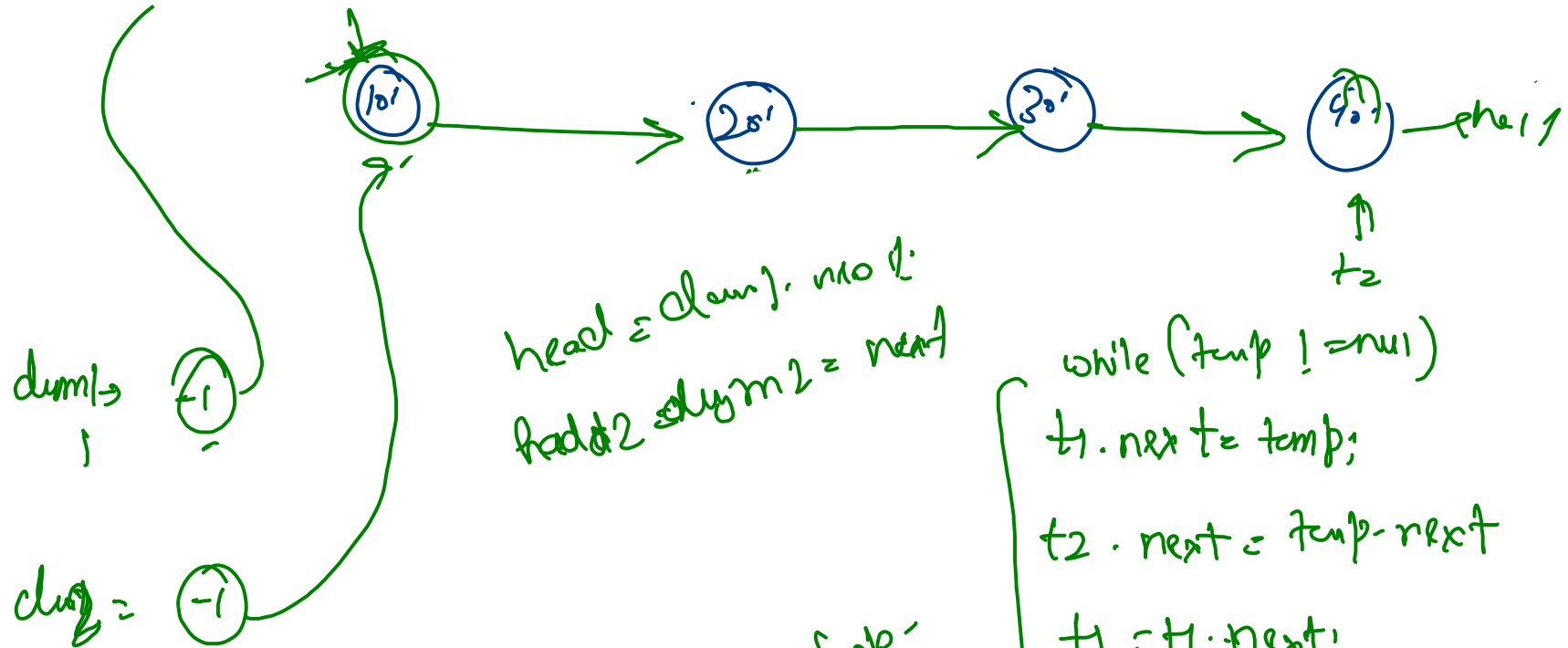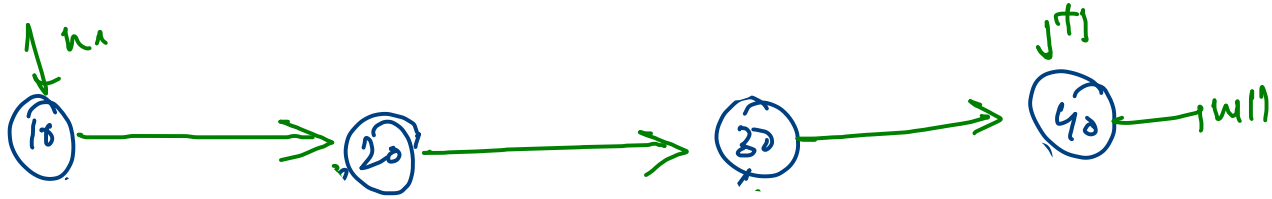   next;

   temp = temp.next.next;

}

Steps-

① Clone linked
   with out setup
   of random point

② connet in zig-zig.

③ Set - random point
   rearrange orlginal.

n

j^{th}

10 ⟶ 20 ⟶ 30 ⟶ 40 ⟶ null

10' ⟶ 20' ⟶ 30' ⟶ 40' ⟶ null

t_2

head = clow]. noo l:
hadd2 dly m2 = next

while (temp != null)
 t1. next = temp;
 t2. next = temp.next
 t1 = t1. next:
 t2 = t2. next.

loop

dum1⟶ -1

cloz = -1

t1. next = null
t2. next = null

temp = temp. next.next